UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA
Richmond Division

ePLUS, INC.,

Plaintiff,

v.

LAWSON SOFTWARE, INC.,

Defendant.

Civil Action No. 3:09-cv-620 (JRS)

DECLARATION OF TODD DOONER

I, Todd Dooner, declare as follows:

1. I make this declaration on my own information, knowledge, and belief.

2. I am a Senior Application Developer for Lawson Software, Inc. My primary duties include developing Lawson's RQC product.

3. Attached hereto as Exhibit 1 is a differences document between the RSS source code and the RQC source code. The differences in the code relate to the version of RQC source code that was released on May 18, 2011.

4. From time to time, Lawson is required to send out patches to the software in order to fix software bugs and update the software for a variety of reasons. Lawson released a patch for the RQC software on June 9, 2011. Attached hereto as Exhibit 2 is a differences document between the RSS source code and the RQC source code for the version of RQC source code that was released on June 9, 2011.

5. Lawson managed the versions of its RSS product and manages its RQC product using Concurrent Versions System ("CVS"). CVS is a control management system that records

the history of source files and documents. A summary of the functionality of CVS is at <http://www.nongnu.org/cvs/>.

6. Attached here to as Exhibit 3 is a true and correct copy of the Lawson CVS files that relate to the RSS and RQC products. The files show the progress of developing RQC, including changes related to:

a. The removal of the intermediate "my cart" feature;

b. Restrictions placed on Punchout sessions so that users cannot place items from more than one punchout vendor on a single requisition;, and

c. Restrictions placed on requisitions to prevent items from item master being placed on the same requisition as punchout items.

7. These documents show the changes are not merely cosmetic, but relate to the functionality of the product.

8. The functional differences between the RQC and RSS product can also be seen based on a comparison of the item load times. Attached hereto as Exhibit 4 are true and correct copies of Fiddler session files that relate to the RSS and RQC products. Fiddler is an HTTP debugging proxy that logs all HTTP traffic between a computer and the Internet. The program allows a user to determine the load speeds for a particular file transfer. A more detailed explanation of the software is located at <http://msdn.microsoft.com/en-us/library/Bb250446.aspx>

9. The Fiddler files show that there are processing differences between RSS and RQC. With RSS, selection of an item from search results causes data regarding the item to be sent to a temporary, or intermediate, list that was referred to as "my cart." The "my cart" list held the item data in a temporary state until the user was ready to create a requisition. Because

the item data was held in a temporary list, the system did not need to conduct error checks when placing an item into the "my cart." Thus, items could be added, deleted, or modified within the "my cart" location without spending time to correct errors.

10. With the RQC product, a user conducts a search and selects a product. Item data for the selected product goes directly to the requisition and all error checks occur immediately. The processing differs from the RSS product that did not require and did not check for errors when moving item data from search results to the "my cart." The direct to requisition functionality is the same as Lawson's Requisitions product. Because error checking must occur before item data is sent to the requisition, the processing steps are different in RQC and RSS.

11. I made the functional code changes to the RQC product. The change from RSS to RQC was not merely cosmetic. Rather, the process by which to requisition was changed to eliminate the intermediate step of creating a temporary list.

12. In designing the RQC product, we were mindful the customers who are using the RQC product are used to seeing items appear on the right of the screen. Therefore, we designed the RQC product to allow the immediately requisitioned items to be mirrored in the same location where the temporary list had been. That creates less interruption to the end user, while fundamentally altering the actual process of doing the requisition.

13. We were also mindful of Lawson's Requisitions ("RQ") product when we developed the RQC product. With the Requisitions product, a selected item was not placed in a temporary list like was done with RSS. When RSS was developed in the late 1990s, the server speed was slower than it is today, and we did not want to force the user to check and clear errors for each product before it was added to the "my cart." To speed things up, we created a temporary list, which upon checking out, would be checked for errors. With RQC, we went back

to the way we used to do things with the RQ process that sends items directly to the requisition and checks for error right away. Because of the increase in server speed caused by overall progress and improvement in computer hardware in 2011, the error checking process under RQC does not unduly delay the process.

I declare under the penalty of perjury the foregoing is true and correct to the best of my knowledge.

Date: September 19, 2011

Todd Dooner